

MPIIO process distribution ...

wputman 22 posts since

Aug 16, 2007 I've got MPI-IO working for FV restarts, but... with a 2-d decomposition the blocks are distributed in a different order from my MPI distribution by the model. As an example, for 12-PEs with an NX=2 by NY=6 decomposition MPI-IO is distributing as:

05 11
04 10
03 09
02 08
01 07
00 06

what I want is this:

10 11
08 09
06 07
04 05
02 03
00 01

Here is the src to do the MPIIO (npx,npy*6) is the size of the global domain, (npes_x, npes_y*6) is my domain decomposition, (is:ie,js:je) is my local domain:

```
call  
MPI_FILE_OPEN(MPI_COMM_WORLD,  
'fvcore_internal_restart',  
MPI_MODE_RDONLY, MPI_INFO_NULL, IUNIT, STATUS)  
VERIFY_(STATUS)  
gsizes(1) = (npx-1)  
gsizes(2) = (npy-1) * 6  
distribs(1) = MPI_DISTRIBUTE_BLOCK  
distribs(2) = MPI_DISTRIBUTE_BLOCK  
dargs(1) = MPI_DISTRIBUTE_DFLT_DARG  
dargs(2) = MPI_DISTRIBUTE_DFLT_DARG  
psizes(1) = npes_x  
psizes(2) = npes_y * 6  
call MPI_COMM_SIZE(MPI_COMM_WORLD, total_pes, STATUS)  
VERIFY_(STATUS)  
call MPI_COMM_RANK(MPI_COMM_WORLD, rank, STATUS)  
VERIFY_(STATUS)  
call MPI_TYPE_CREATE_DARRAY(total_pes, rank, 2, gsizes, distribs,  
dargs, psizes, MPI_ORDER_FORTRAN, MPI_DOUBLE_PRECISION, filetype,  
STATUS)
```

MPIIO process distribution ...

```
VERIFY_(STATUS)
call MPI_TYPE_COMMIT(filetype, STATUS)
VERIFY_(STATUS)
lsize = (ie-is+1)*(je-js+1)
offset = 1248 ! sequential access: 4 + INT(6) + 8 + INT(5) + 8 +
DBL(NPZ+1) + 8 + DBL(NPZ+1) + 8
           !                         4 + 24      + 8 + 20      + 8 +
584       + 8 + 584      + 8 = 1248
! U

do
k=0,npz-1
  call MPI_FILE_SET_VIEW(IUNIT, offset, MPI_DOUBLE_PRECISION,
filetype,
"native", MPI_INFO_NULL, STATUS)
  VERIFY_(STATUS)
  call MPI_FILE_READ_ALL(IUNIT, U(:,:,k), lsize,
MPI_DOUBLE_PRECISION, mpistatus, STATUS)
  VERIFY_(STATUS)
  offset = offset + (npx-1)*(npy-1)*ntiles*8 + 8
enddo
! V

do
k=0,npz-1
  call MPI_FILE_SET_VIEW(IUNIT, offset, MPI_DOUBLE_PRECISION,
filetype,
"native", MPI_INFO_NULL, STATUS)
  VERIFY_(STATUS)
  call MPI_FILE_READ_ALL(IUNIT, V(:,:,k), lsize,
MPI_DOUBLE_PRECISION, mpistatus, STATUS)
  VERIFY_(STATUS)
  offset = offset + (npx-1)*(npy-1)*ntiles*8 + 8
enddo
! PT

do
k=0,npz-1
  call MPI_FILE_SET_VIEW(IUNIT, offset, MPI_DOUBLE_PRECISION,
filetype,
"native", MPI_INFO_NULL, STATUS)
  VERIFY_(STATUS)
  call MPI_FILE_READ_ALL(IUNIT, PT(:,:,k), lsize,
MPI_DOUBLE_PRECISION, mpistatus, STATUS)
  VERIFY_(STATUS)
  offset = offset + (npx-1)*(npy-1)*ntiles*8 + 8
enddo
! PE

do
k=0,npz
  call MPI_FILE_SET_VIEW(IUNIT, offset, MPI_DOUBLE_PRECISION,
filetype,
```

MPIIO process distribution ...

```
"native", MPI_INFO_NULL, STATUS)
VERIFY_(STATUS)
call MPI_FILE_READ_ALL(IUNIT, PE(:,:,k), lsize,
MPI_DOUBLE_PRECISION, mpistatus, STATUS)
VERIFY_(STATUS)
offset = offset + (npx-1)*(npy-1)*ntiles*8 + 8
enddo
! PKZ

do
k=0,npz-1
call MPI_FILE_SET_VIEW(IUNIT, offset, MPI_DOUBLE_PRECISION,
filetype,
"native", MPI_INFO_NULL, STATUS)
VERIFY_(STATUS)
call MPI_FILE_READ_ALL(IUNIT, PKZ(:,:,k), lsize,
MPI_DOUBLE_PRECISION, mpistatus, STATUS)
VERIFY_(STATUS)
offset = offset + (npx-1)*(npy-1)*ntiles*8 + 8
enddo
call MPI_FILE_CLOSE(IUNIT, STATUS)
VERIFY_(STATUS)
```

Tags: fortran, mpi, mpiio, i/o

[wputman](#) 22 posts since

Aug 16, 2007 [1. Re: MPIIO process distribution](#) Apr 13, 2009 3:31 PM

I think I have it now, I just use the row,col number to create an mpiio_rank number that is how MPIIO layouts out the pes and decides where to write, it validates for my 2x6 and 3x18 tests. Here is my code to create the MPIIO rank:

```
mcol = npes_x
mrow = npes_y*6
irow = rank/mcol      !! logical row number
jcol = mod(rank, mcol) !! logical column number
mpiio_rank = jcol*mrow + irow

call MPI_TYPE_CREATE_DARRAY(total_pes, mpiio_rank, 2, gsizes,
distrib, dargs, &
& psizes, MPI_ORDER_FORTRAN, MPI_DOUBLE_PRECISION, filetype,
STATUS)
```

This doesn't work if my decomposition is not evenly divisible, ie 16x6 for a c360 grid where 360 doesn't evenly divide by 16.

[oloso](#) 6 posts since

Aug 16, 2007 [2. Re: MPIIO process distribution](#) Apr 14, 2009 6:11 PM

in response to: [wputman](#) Trying to follow you. Are you saying you have 360 grid points in the Y-direction that you are trying to distribute among 16x6 processes?